DOCKET: FIS919990319US1 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTOR:	Harry J. Beatty et al.	EXAMINER:	Ali, S.J.
SERIAL NO.:	09/597,524	ART UNIT:	2127
FILING DATE:	June 20, 2000	DATE:	October 19, 2005
FOR:	Method of Using a Distinct Flow of Computational Control as a Reusable Abstract Data Object		

SECOND DECLARATION UNDER RULE 131 OF HARRY J. BEATTY AND PETER ELMENDORF

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

We, Peter Elmendorf and Harry J. Beatty III, do hereby declare as follows:

1. We are employees of International Business Machines Corporation ("IBM"), the assignee of the above-identified patent application entitled "Method of Using a Distinct Flow of Computational Control as a Reusable Abstract Data Object." We are co-inventors of the invention described and claimed in the above-identified application.

2. This is a declaration under the provisions of 37 CFR § 1.131 for the purpose of swearing back of a reference which was cited in the subject application, and supplements our previous declaration dated May 10, 2005. This declaration establishes facts showing conception and reduction to practice of this invention in this country prior to

the September 29, 1999 filing date of Sievert et al. U.S. Patent No. 6,832,376 cited against this application, and due diligence from a time prior to that date until the application was filed.

3. The claimed invention in the above-identified application was conceived by us in the United States prior to September 29, 1999. This is evidenced by the copy of a portion of an invention disclosure form created by Peter Elmendorf on our behalf describing the invention disclosed in the subject patent application, attached hereto as Exhibit A. The Exhibit A disclosure references an electronic document created by us "bottle.prz," which is the drawing attached as Exhibit B.

4. The disclosure of Exhibit A was created, and is dated, prior to September 29, 1999, the filing date of the Sievert et al. '376 patent. The drawing of Exhibit B was also created prior to December 20, 1999. Actual dates and the material not pertinent to conception and reduction to practice of the invention have been redacted in view of their confidential nature.

5. The invention claimed in the subject application is directed to a method of parallel processing in a memory structure employing first and second threads, with the first thread waiting for and processing work prepared for it by the second thread. The Exhibit A disclosure uses the term "first thread" in the same manner as used in the claims of the subject application, while the Exhibit B drawing refers to the first thread as "thread (in a bottle)", using the same "bottle" terminology for the first thread as used in column 12, lines 24-26 and Fig. 11 of the subject application. However, the Exhibit A disclosure and Exhibit B drawing use different terms for the "second thread" referenced in the claims; the

Exhibit A disclosure uses the term "software" and the Exhibit B disclosure uses the term "escapement" for what is described in the claims as the "second thread," and which is also described in the subject application as the "launcher." See column 12, lines 52-56 and Fig. 11. Although the concept is the same, in preparing the subject application we decided to use the term "launcher" instead of the term "escapement" since the former would be more widely understood.[1] Thus, the Exhibit B drawing shows the actions of the claimed first thread in the left side flow chart labeled "Create Thread (in a bottle)" and the actions of the claimed second thread in the center flow chart labeled "escapement (work, fcn)."

6. The following chart compares the steps of the claimed method in detail with the teachings in the original disclosure and drawing of Exhibits A and B:

CLAIM 1	ORIGINAL DISCLOSURE
A method of parallel processing in a memory structure comprising	"parallel programming" p.1, Ex. A; "new paradigm for the use of threads in a parallel environment" p.2, Ex. A.
creating a first thread in the memory structure which represents an independent flow of control managed by a program structure,	"Threads are data objects." p.2, Ex. A; "The invention implements an abstract data object which has a first thread waiting on it." p.2, Ex. A; "Create Thread (in a bottle)" Ex. B.

[1] The term "escapement" is defined as "a device in a timepiece which controls the motion of the train of wheelwork and through which the energy of the power source is delivered to the pendulum or balance by means of impulses that permit a tooth to escape from a pallet at regular intervals." Merriam-Webster OnLine Dictionary, www.m-w.com. Thus, an "escapement" is analogous to the "launcher" or "second thread" which controls the pace of the first thread by assigning it work.

said first thread having two states, a first state processing work for the program structure and a second state undispatched awaiting work to process;	"When desired, the software [second thread] assigns particular work to the data object, which the waiting [first] thread then wakes up and does. After performing the work, the [first] thread again waits for more work." p.2, Ex. A;
providing a second thread in the memory structure which represents an independent flow of control managed by a program structure separate from the first thread;	"escapement (work, fcn)" Ex. B.
using the second thread to prepare work for the first thread to process;	"When desired, the software [second thread] assigns particular work to the data object, which the waiting thread then wakes up and does." p.2, Ex. A; In the second thread "escapement" flow chart, the steps "Inject data." Ex. B.
placing the work prepared by the second thread in a queue for processing by the first thread;	"When desired, the software [second thread] assigns particular work to the data object, which the waiting thread then wakes up and does." p.2, Ex. A; In the second thread "escapement" flow chart, the steps "Unlock bottle [first thread] Wait for escapement [second thread] release." Ex. B.
if the first thread is awaiting work to process when the work prepared by the second thread is placed in the queue, dispatching the first thread and using it to process the work in the queue;	"When desired, the software [second thread] assigns particular work to the data object, which the waiting thread then wakes up and does." p.2, Ex. A; In the first thread "Thread (in a bottle)" flow chart, the steps "Lock 'waiting for results' Load data Release escapement [second thread] Work on data Unlock 'waiting for results'." Ex B.

if the first thread is processing other work when the work prepared by the second thread is placed in the queue, using the first thread to complete processing of the other work, access the work in the queue, and then process the work in the queue; and	"After performing the work, the [first] thread again waits for more work." p.2, Ex. A; "Threads are created once and reused as needed." p.2, Ex. A; In the first thread "Thread (in a bottle)" flow chart, the steps "Lock 'waiting for results' Load data Release escapement [second thread] Work on data Unlock 'waiting for results'." Ex B.
using the program structure to destroy the first thread in the memory structure after the first thread completes a desired amount of work.	"Threads are created once and reused as needed." p.2, Ex. A; "The [first] thread is not destroyed until the application program decides to do so." p.2, Ex. A.

7. The disclosure and drawing of Exhibits A and B, respectively, were submitted to IBM's patent attorneys prior to September 29, 1999.

8. On November 11, 1999, we received from Jay Anderson, the IBM patent attorney responsible for the subject patent application, a communication reporting results of a search made to determine the patentability of the invention disclosed in our invention disclosure attached as Exhibit A and Exhibit B. We subsequently provided our comments to Mr. Anderson regarding the search results.

9. We also reduced to practice the invention described in the disclosure and drawing of Exhibits A and B, respectively, in the United States prior to September 29, 1999. This reduction to practice was a "working implementation" for a "compiler product" as noted on page 3 of Exhibit A. The reduction to practice implemented all of the steps and limitations described in claim 1 of the subject patent application.

10. In addition to the reduction to practice, we worked diligently with patent counsel from a time before the date of the Sievert '376 patent reference, September 29,

1999, until the June 20, 2000 filing date of the subject application. Our work with IBM's outside patent counsel on the subject application is described in our previous declaration dated May 10, 2005, the contents of which are hereby incorporated by reference.

11. We declare further that all statements made herein on information and belief are believed to be true; and further that these statements and the like so made are punishable by fine or imprisonment or both, under §1001 of Title XVIII of the United States Code and that such willful false statement may jeopardize the validity of the application or any patent issuing thereon.

Peter Elmendorf — Date: Oct 19, 2005

Harry J. Beaty III — Date: 10/19/2005

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date indicated below as first class mail in an envelope addressed to Mail Stop _____, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Name: Barbara Browne Date: 11/1/2005 Signature:

EXHIBIT A

[redacted] Method of using a distinct flow of [redacted] ational control as a reusable abstract data objec[redacted] [redacted]

Disclosure FIS8-1999-0335

Created By: Peter Elmendorf **Created On:** [redacted]
Last Modified By: [redacted] **Last Modified On:** [redacted]

*** IBM Confidential ***

Required fields are marked with the asterisk (*) and must be filled in to complete the form .

Summary

Status	Under Evaluation
Processing Location	FIS
Functional Area	GBF Software Related Services, Applications and Solutions ... 600 ACA
Attorney/Patent Professional	Jay Anderson/Fishkill/IBM
IDT Team	[redacted]
Submitted Date	[redacted]
Owning Division	[redacted]
PVT Score	[redacted]

Inventors with Lotus Notes IDs

Inventors: Peter Elmendorf/Fishkill/IBM, Harry Beatty III/Fishkill/IBM

Inventor Name > denotes primary contact	Inventor Serial	Div/Dept	Manager Serial	Manager Name
[redacted] > Elmendorf, Peter C [redacted]	[redacted]	[redacted]	[redacted]	[redacted]
Beatty III, Harry J				

Inventors without Lotus Notes IDs

IDT Selection

IDT Team:	Attorney/Patent Professional:
[redacted]	Jay Anderson/Fishkill/IBM

Response Due to IP&L [redacted]

Main Idea

Title of disclosure (in English)

Method of using a distinct flow of computational control as a reusable abstract data object.

Idea of disclosure

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

In parallel programming, it is customary to use one or more threads within a process. Each thread is assigned a specific
unit of work to perform,
generally in parallel, and when the work is finished, the threads cease to exist. There is a cost to create a thread, terminate

[redacted]

a thread, and to manage a thread. The cost has both machine-cycle components and programming complexity components. The programming complexity components are a source of errors in implementation and design of the software.

The prevailing paradigm in the use of threads treats the threads and data differently. There is control flow (threads), and there is data. The resulting dichotomy creates an environment which tends to place fetters on the kinds of solutions envisioned, and creates
complexity and resulting error-proneness during implementation.

The invention implements a new paradigm for the use of threads in a parallel environment. The invention essentially creates a
thread and captures it, binding it to a data object which, from the programmer's perspective, is abstract. This allows a thread
(or flow of control) to be treated as a data object by the software.

This has a number of advantages.

1. Threads are created once and reused as needed. This avoids thread creation and destruction costs found in prevailing approaches.

2. Threads are data objects. This eliminates the prevailing dichotomy between control and data, giving a programmer a greater
mental field on which to envision solutions to problems.

3. Because threads are data objects, previously impossible operations are available to a software developer.
These operations include, but are not limited to, attaching threads to other data objects (for later use in execution),
passing threads (control flows) as parameters, etc.


bottle.prz

2. How does the invention solve the problem or achieve an advantage,(a description of "the invention", including figures inline as appropriate)?
The invention implements an abstract data object which has a thread waiting on it. The data object can be passed around
and incorporated into the data structures of a program, as can any traditional data object. When desired, the software assigns
particular work to the data object, which the waiting thread then wakes up and does. After performing the work, the thread
again waits for more work. The work may be assigned from any section of the application, at any desired time. The thread is not destroyed until the application program decides to do so.

This approach greatly simplifies the creation of software that needs to leverage parallel operation by use of threads. By abstracting
the thread, burdensome details are removed from the purview of the programmer. By encapsulating a waiting thread as a
data object, the programmer has more degrees of freedom and greater range of solutions, because the dichotomy between
control flow and data is bridged.

3. If the same advantage or problem has been identified by others (inside/outside IBM), how have those

[redacted]

others solved it and does your solution differ and why is it better?
I am not aware of anything like this.

4. If the invention is implemented in a product or prototype, include technical details, purpose, disclosure details to others and the date of that implementation.
There is already a working implemenation. It is intended to be placed in the DCL compiler product, release date[redacted]
however, beta release may occur much earlie[redacted]

***Critical Questions (Questions 1 - 7 must be answered)**

***Question 1**
On what date was the invention workable? [redacted] **Please format the date as MM/DD/YYYY**
(Workable means i.e. when you know that your design will solve the problem)

***Question 2**
Is there any planned or actual publication or disclosure of your invention to anyone outside IBM? [redacted] Yes No

If yes, Enter the name of each publication or patent and the date published below.
Publication/Patent:
Date Published or Issued:

Are you aware of any publications, products or patents that relate to this invention? [redacted] Yes No

If yes, Enter the name of each publication or patent and the date published below.
Publication/Patent:
Date Published or Issued:

***Question 3**
Has the subject matter of the invention or a product incorporating the invention been sold, used [redacted] Yes No
internally in manufacturing, announced for sale, or included in a proposal?

Is a sale, use in manufacturing, product announcement, or proposal planned? [redacted] Yes No

If Yes, identify the product if known and indicate the date or planned date of sale, announcements, or proposal and to whom the sale, announcement or proposal has been or will be made.
Product: [redacted]
Version/Release: [redacted]
Code Name:
Date: [redacted]
To Whom: [redacted]
If more than one, use cut and paste and append as necessary in the field provided.

***Question 4**
Was the subject matter of your invention or a product incorporating your invention used in [redacted] Yes No
public, e.g., outside IBM or in the presence of non-IBMers?
If yes, give a date. **Please format the date as MM/DD/YYYY**

***Question 5**
Have you ever discussed your invention with others not employed at IBM? [redacted] Yes No
If yes, identify individuals and date discussed. Fill in the text area with the following information: the names of the individuals, the employer, date discussed, under CDA, and CDA #

[redacted]

Question 6	
Was the invention, in any way, started or developed under a government contract or project?	[redacted]
If Yes, enter the contract number	

*Question 7	
Was the invention made in the course of any alliance, joint development or other contract activities?	[redacted]
If Yes, enter the following: Name of Alliance, Contractor or Joint Developer	
Contract ID number	
Relationship contact name	
Relationship contact E-mail	
Relationship contact phone	

Question 8	
Have you submitted, or are you aware of, any related disclosure submission?	[redacted]
If Yes, please provide the title and docket or disclosure number below:	

Question 9

What type of companies do you expect to compete with inventions of this type? *Check all that apply*

- [redacted] Manufacturers of enterprise servers
- Manufacturers of entry servers
- Manufacturers of workstations
- Manufacturers of PC's
- Non-computer manufacturers
- Developers of operating systems
- Developers of networking software
- Developers of application software
- Integrated solution providers
- Service providers
- Other (Please specify below):

Patent Value Tool (Optional - this may be used by the inventor and attorney to assist with the evalua

(The Patent Value tool can be used by you or the evaluation team to determine the potential licensing value of your invention.)

These are the answers which were entered into the **Patent Value Tool.**

Market

What is the anticipated annual market size (in dollars) that will be captured by your invention?

[redacted]

Question 1 - How new is the technical field?

[redacted]

Question 2 - How central is the invention to the product(s) which might be expected to contain the invention?

[redacted]

Reason(s) for above Answer [redacted]

[redacted]

Question 3 - What is the scope of the claim?

[redacted]

PORTFOLIO NEED View PPM Needs List

What are the portfolio needs in the area of your invention?

[redacted]

EXPLOITATION & ENFORCEMENT

Question 1 - How easily can the use of the invention by a competitor be detected?

[redacted]

Question 2 - How easily can the use of the invention be avoided by a competitor?

[redacted]

BUSINESS VALUE

Question 1 - What percentage of the companies producing products in the field of this invention might use this invention?

[redacted]

Reason(s) for above Answer [redacted]g

[redacted]

Question 2 - What is the value of this patent to current or anticipated Alliance Activity between IBM and other companies?

[redacted]

Reason(s) for above Answer [redacted]

[redacted]

Question 3 - What is the value of this patent to current or anticipated Technology Transfer Activity between IBM and other companies?

[redacted]

Reason(s) for above Answer [redacted]s

[redacted].

Question 4 - Does it result in prestige to IBM?

[redacted]

Reason(s) for above Answer [redacted]

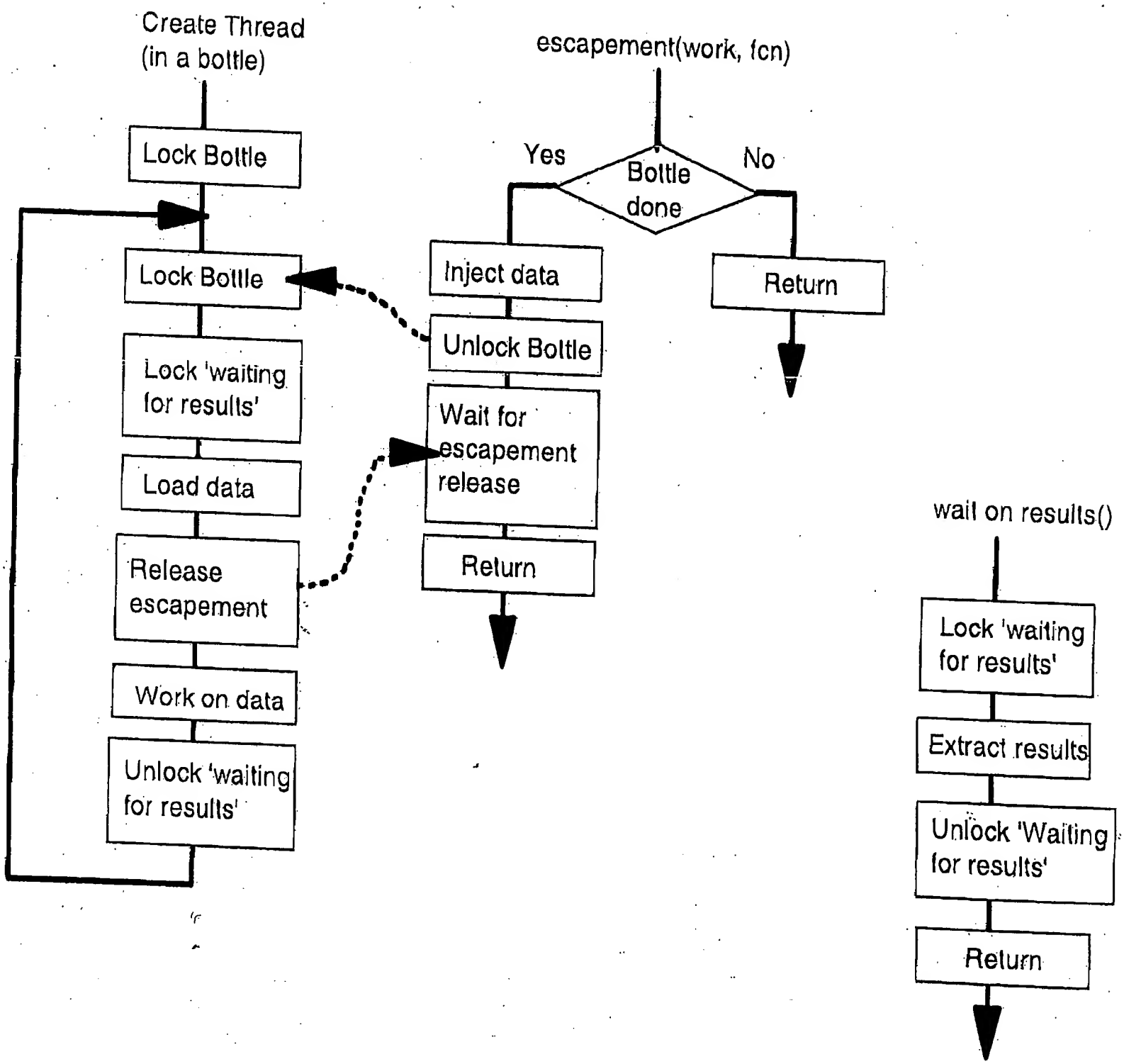
[redacted]

Post Disclosure Text & Drawings

Enter any additional information relating to this disclosure below:

(Form Revised 12/17/97)

EXHIBIT B


Create Thread
(in a bottle)
Lock Bottle
Lock Bottle
Lock 'waiting
for results'
Load data
Release
escapement
Work on data
Unlock 'waiting
for results'
escapement(work, fcn)
Yes
Bottle
done
No
Inject data
Unlock Bottle
Wait for
escapement
release
Return
Return
wait on results()
Lock 'waiting
for results'
Extract results
Unlock 'Waiting
for results'
Return